

[Home \(/\)](#)[Open-Source \(/open-source\)](/open-source)[Blog \(/blog\)](/blog)[Link \(/link\)](/link)[About \(/about\)](/about)

# Spring Data JPA 入门

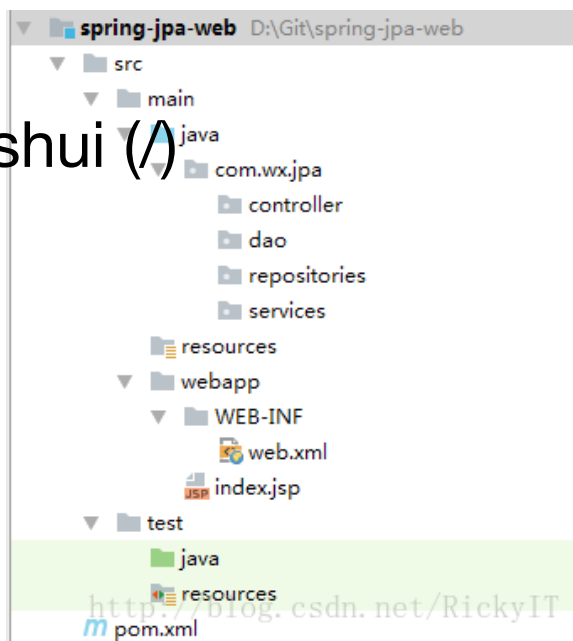
📅 2017/01/12

项目中有用到spring data jpa, 简单看了一下, 在基础的数据库操作上, 要比mybatis好用, 简化了持久层的操作, 使coder可以专注于业务逻辑的实现。网上参考了一些文章, 实际coding的时候还是出好多错误。这里通过CRUD, 记录摸索填坑的过程。详细的概念可以参考文末的文章。

## 创建工程




创建一个maven工程。



## 配置

### web.xml



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <display-name>spring-jpa-web</display-name>
  <!-- spring mvc -->
  <servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>classpath:spring.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>appServlet</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>

</web-app>
```

## pom.xml

特别说明，spring-data-jpa要用版本1.9.0.RELEASE。



# Wuxinshui

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>org.wxs.spring</groupId>
  <artifactId>spring-jpa-web</artifactId>
  <packaging>war</packaging>
  <version>0.1.0</version>
```

```
  <properties>
    <spring_version>4.3.0.RELEASE</spring_version>
    <spring_jpa_version>1.9.0.RELEASE</spring_jpa_version>
    <spring_data_commons_version>1.12.6.RELEASE</spring_data_commons_version>
    <hibernate.version>4.2.5.Final</hibernate.version>
  </properties>
```

```
  <dependencies>
```

```
    <!-- PERSISTENCE begin -->
```

```
    <!--spring data jpa start-->
```

```
    <dependency>
```

```
      <groupId>org.springframework.data</groupId>
```

```
      <artifactId>spring-data-jpa</artifactId>
```

```
      <version>${spring_jpa_version}</version>
```

```
    </dependency>
```

```
    <!-- hibernate -->
```

```
    <dependency>
```

```
      <groupId>org.hibernate</groupId>
```

```
      <artifactId>hibernate-core</artifactId>
```

```
      <version>${hibernate.version}</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.hibernate</groupId>
```

```
      <artifactId>hibernate-entitymanager</artifactId>
```

```
      <version>${hibernate.version}</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.hibernate</groupId>
```

```
      <artifactId>hibernate-ehcache</artifactId>
```

```
      <version>${hibernate.version}</version>
```

```
    </dependency>
```



```
<!--连接池-->
<dependency>
  <groupId>org.apache.tomcat</groupId>
  <artifactId>tomcat-jdbc</artifactId>
  <version>7.0.54</version>
</dependency>

<!--mysql-->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.35</version>
</dependency>

<!-- spring start -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-web</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>${spring_version}</version>
</dependency>
<dependency>
```



```

    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>${spring_version}</version>
  </dependency>
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${spring_version}</version>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
    <version>${spring_version}</version>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${spring_version}</version>
  </dependency>
  <!-- spring end -->
  <!-- test -->
  <dependency>
    <!-- 使用junit4-->
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <!-- 3.0是使用编程的方式-->
    <!--<version>3.8.1</version>-->
    <!-- 4.0使用注解的方式-->
    <!--SpringJUnit4ClassRunner requires JUnit 4.12 or higher-->
    <version>4.12</version>
    <scope>test</scope>
  </dependency>

  <!-- servlet start -->
  <dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>tomcat-servlet-api</artifactId>
    <version>8.0.3</version>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>org.apache.tomcat</groupId>
    <artifactId>jsp-api</artifactId>

```



```
<version>6.0.39</version>
<scope>provided</scope>
</dependency>
<dependency>
  <groupId>jstl</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>

<dependency>
  <groupId>javax.servlet.jsp</groupId>
  <artifactId>jsp-api</artifactId>
  <version>2.1</version>
  <scope>provided</scope>
</dependency>
<!-- servlet end -->
</dependencies>

</project>
```

## spring.xml

 Wuxinshui

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:jpa="http://www.springframework.org/schema/data/jpa"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
    http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.0.xsd
    http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd
    http://www.springframework.org/schema/data/jpa http://www.springframework.org/schema/data/jpa/spring-jpa-1.3.xsd"
    default-lazy-init="true">

<description>Spring公共配置</description>

<!--service 包-->
<context:component-scan base-package="com.wx.jpa"/>

<!-- Jpa Entity Manager 配置 -->
<bean id="entityManagerFactory" class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
    <property name="dataSource" ref="dataSource"/>
    <property name="jpaVendorAdapter" ref="hibernateJpaVendorAdapter"/>
    <property name="packagesToScan" value="com.wx.jpa.entities"/>
    <property name="jpaProperties">
        <props>
            <!-- 命名规则 My_NAME->MyName -->
            <prop key="hibernate.ejb.naming_strategy">org.hibernate.cfg.ImprovedNamingStrategy</prop>
        </props>
    </property>

    <property name="jpaDialect">
        <bean class="org.springframework.orm.jpa.vendor.HibernateJpaDialect"/>
    </property>
</bean>

<bean id="hibernateJpaVendorAdapter" class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
    <property name="generateDdl" value="true"/>
    <property name="database" value="MYSQL"/>
    <property name="databasePlatform" value="org.hibernate.dialect.MySQLInnoDBDialect"/>
</bean>

<!-- Spring Data Jpa 配置 -->
<jpa:repositories base-package="com.wx.jpa.repository" transaction-manager-ref="transactionManager"
    entity-manager-factory-ref="entityManagerFactory"/>

<!-- Jpa 事务配置 -->
<bean id="transactionManager" class="org.springframework.orm.jpa.JpaTransactionManager">
```



```
<property name="entityManagerFactory" ref="entityManagerFactory"/>
</bean>
```



```
<!-- 使用annotation定义事务 -->
<tx:annotation-driven transaction-manager="transactionManager" proxy-target-class="true"/>
```

```
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="dataSource"/>
</bean>
```

```
<!-- 数据源配置, 使用Tomcat JDBC连接池 -->
```

```
<bean id="dataSource" class="org.apache.tomcat.jdbc.pool.DataSource" destroy-method="close">
```

```
    <!-- Connection Info -->
```

```
    <property name="driverClassName" value="{jdbc.driver}"/>
```

```
    <property name="url" value="{jdbc.url}"/>
```

```
    <property name="username" value="{jdbc.username}"/>
```

```
    <property name="password" value="{jdbc.password}"/>
```

```
    <!-- Connection Pooling Info -->
```

```
    <property name="maxActive" value="{jdbc.pool.maxActive}"/>
```

```
    <property name="maxIdle" value="{jdbc.pool.maxIdle}"/>
```

```
    <property name="minIdle" value="0"/>
```

```
    <property name="defaultAutoCommit" value="true"/>
```

```
</bean>
```

```
<!-- 数据库连接-->
```

```
<context:property-placeholder location="classpath:jdbc.properties"/>
```

```
</beans>
```

## 属性文件


jdbc.properties

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/seckill?useUnicode=true&characterEncoding=utf8
jdbc.username=root
jdbc.password=admin
jdbc.pool.maxIdle=5
jdbc.pool.maxActive=10
```

# Entity

## Wuxinshui (/)

新建表 user

名	类型	长度	小数点	不是 null	
▶ id	int	11	0	<input checked="" type="checkbox"/>	 1
name	varchar	32	0	<input type="checkbox"/>	
password	varchar	32	0	<input type="checkbox"/>	
birthday	date	0	0	<input type="checkbox"/>	

默认:

注释:  

☒ 自动递增


☐ 无符号

☐ 填充零

<http://blog.csdn.net/RickyIT>

## 实体类

新建实体类User



```
@Entity
@Table(name = "user")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    @Column(name = "name")
    private String name;
    @Column(name = "password")
    private String password;
    @Column(name = "birthday")
    private Date birthday;

    /**
     * The default constructor only exists for the sake of JPA. You won't use it directly, so it is designated as protected.
     */
    protected User() {
    }

    public User(String name, String password, Date birthday) {
        this.name = name;
        this.password = password;
        this.birthday = birthday;
    }
}
```

getter、setter省略

## DAO层

创建接口UserRepository

```
public interface UserRepository extends JpaRepository<User,Integer> {

}
```

## 测试

## 新增



@Test

```
public void test1() {  
    User user=new User("SL1","123",new Date());  
    userRepository.save(user);  
    System.out.println("save success");  
}
```

查询创建工具 查询编辑器

```
1 SELECT * FROM `user` WHERE `name`='SL1'
```

信息 结果1 概况 状态

id	name	password	birthday
9	SL1	123	2017-01-12

+ - ✓ ✕ ↺ ⌂

```
SELECT * FROM `user` WHERE `name`='SL1'
```

一月 12, 2017 5:07:43 下午 org.hibernate.tool.hbm2ddl.SchemaUpd  
INFO: HHH0000232: Schema update complete  
save success


<http://blog.csdn.net/RickyIT>

## 查询

### 查询全部

```
@Test  
public void test2(){  
    for (User user:userRepository.findAll()) {  
        System.out.println(user);  
    }  
    System.out.println("search all success");  
}
```

### 查询结果



```
User{id=9, name='SL1', password='123', birthday=2017-01-12 00:00:00.0}
User{id=10, name='SL1', password='123', birthday=2017-01-12 00:00:00.0}
User{id=11, name='SL1', password='123', birthday=2017-01-13 00:00:00.0}
search one success
```

## 根据ID查询

```
@Test
public void test3(){
    System.out.println(userRepository.findOne(9));
    System.out.println("search one success");
}
```

### 查询结果

```
User{id=9, name='SL1', password='123', birthday=2017-01-12 00:00:00.0}
search one success
```

## 更新

如果有Id, jpa会认为是更新, 没有则是新增。

```
@Test
public void test4(){
    User user=new User("SL99","123",new Date());
    user.setId(9);
    userRepository.save(user);
    test3();
    System.out.println("update success");
}
```

### 更新成功

```
User{id=9, name='SL99', password='123', birthday=2017-01-13 00:00:00.0}
search one success
update success
```

## 删除



```
@Test
public void tests() {
    userRepository.delete(9);
    test3();
    System.out.println("delete success");
}
```

删除成功

```
null
search one success
delete success
```

## 踩过的坑

### 坑1

```
java.lang.AbstractMethodError: org.springframework.data.repository.core.support.RepositoryFactorySupport.getTargetRepository(Lorg/springf
```

**解决方案** (<http://stackoverflow.com/questions/32405886/spring-data-rest-pagingandsortingrepository-abstractmethoderror-repositoryfacto>)

Deleting the spring-data-commons and changing the spring-data-jpa version to 1.9.0.RELEASE like @peeskillet said now works fine! 这也是为什么pom文件中spring-data-jpa要配置1.9.0.RELEASE的版本。

### 坑2

```
java.lang.IllegalArgumentException: Not an managed type: class java.lang.Object
```

**解决方案** (<http://stackoverflow.com/questions/14069449/autowiring-fails-not-an-managed-type>)